POLITICAL OPINION MINING FOR POPULARITY PREDICTION

A Mini Project Report

submitted by

AKSHAYKUMAR M R(MES20MCA-2003)

to

the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree

of

Master of Computer Applications

Department of Computer Applications

MES College of Engineering
Kuttippuram, Malappuram - 679 582

February 2022

DECLARATION

I undersigned hereby declare that the project report **POLITICAL OPINION MINING FOR POPULARITY PREDICTION**, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala, is a bonafide work done by me under supervision of Mr.Hyderali K, Associate Professor, Department of Computer Applications. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place:Kuttippuram

Date:04-03-2022

AKSHAYKUMAR M R

DEPARTMENT OF COMPUTER APPLICATIONS
MES COLLEGE OF ENGINEERING, KUTTIPPURAM

CERTIFICATE

This is to certify that the report entitled **POLITICAL OPINION MINING FOR POPULARITY PREDICTION** is a bona fide record of the Mini Project work carried out by **AKSHAYKUMAR M R(MES20MCA-2003)** submitted to the APJ Abdul Kalam Technological University, in partial fulfillment of the requirements for the award of the Master of Computer Applications, under my guidance and supervision. This report in any form has not been submitted to any other University or Institution for any purpose.

Internal Supervisor(s) External Supervisor(s)

Head Of The Department

Acknowledgements

At the very outset I would like to thank the almighty's mercy towards me over the years... I wish to express my sincere thanks to my project guide Ms. Priya J.D, Assistant professor, Dept. of Master of Computer Applications who guided me for the successful completeness of this project. I also thank her for valuable suggestions, guidance, constant encouragement, boundless corporation, constructive comments and motivation extended to me for completion of this project work. I would express my sincere thanks to my internal guide Mr. Hyderali K, Head of Department for their immense guidance to complete the project successfully. I would like to express my sincere thanks to all the faculty members of Computer Applications department for their support and valuable suggestion for doing the project work. Last but not least my graceful thanks to my parents, friends and also the persons who supported me directly and indirectly during the project

AKSHAYKUMAR M R(MES20MCA-2003)

Abstract

This POLITICAL OPINION MINING FOR POPULARITY PREDICTION project is a web application build by machine learning technology.The proliferation of social media in the recent past has provided end users a powerful platform to voice their opinions. One such application is in the field of politics, where political entities need to understand public opinion and thus determine their campaigning strategy. Also by the help of this popularity of a person can be predicted in Politics also which will help the party to understand the sentiment and opinion of public about their party member which can help them in wining a election. Sentiment analysis on social media data has been seen by many as an effective tool to monitor user preferences and inclination. This research proposes an approach that is based on Twitter based Political opinion mining for predicting the popularity of a person on a given set of Tweets containing varied opinion. The objective is to extract expressions of opinion and predict the personality of political member by classifying them as positive or negative.This approach is going to use the concept of Sentiment Analysis i.e. tracking opinion of public, which uses the natural language processing and extract the information like either public's view is positive or negative which can be used further to predict popularity of political party member. The data that is to be taken here is from twitter tweets

Keywords:Tweepy, TextBlob,Naive Bayes

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

1.1.1 Twitter

Twitter a social networking service was launched in 2006 and gained popularity among people with over 500 million registered users. Twitter is in the top ten most used social media applications on the internet. Twitter simply broadcast tweets with 140 characters or less to the followers (one who chooses other's tweets to be posted in their timeline) around the network. These tweets represent any information in text form or shared link relating personal activity, entertainment, sports, or politics.

Nowadays importance of Twitter has risen as it has proved to be fast, low cost medium for disseminating real time information for any sporting event, disaster, academic conference, political activities, elections etc. It was found to be most effectives during some of the worth mentioning events like Michael Jackson's Death 2009, Iranian election protest 2009-10, FIFA World Cup 2010, NBA Finals 2010, Egyptian revolution 2011, Sandy Hurricane 2012 and US Election 2010. During these events twitter users updated their statuses and tweeted thousands of tweets per second and setting up tweet propagation per unit time records. These tweets resulted in revolutions for certain nations, sometimes celebrating one's win, sometimes benefitting people suffering from disasters like earthquake, flooding, and sometimes aiding politician for their campaigns. These events got popularity on the basis of terms, words or phrases referred by users commonly in their tweets, so most frequent used words would rather

help in trend topic setting. These trending topics would quickly suggest for a user "what is happening right now". These trending topics mostly range between real-time events. People send their reactions, opinions, endorse someone else ideas and discuss about these events in the form of tweets. These status updates are reflected on user's page and also in the timeline of follower's pages. Direct messages can be sent to targeted users using '@username' for one-to-one correspondence. Tweets starting by 'RT' is indicator for retweet in which user endorse and propagate someone else's interesting post to rest of his followers. These tweet statuses and re-tweets will form trends if more frequently discussed over twitter.

1.1.2 Social Media Politics

Campaigns for elections have found totally new platform provided by different social media applications. The popularity rise can be studied for developed and under developed countries. Political parties, party workers, and politician post their campaign messages over fan pages and twitter accounts, maintained by themselves or paid party employees. Number of followers reflects the popularity of political figure and agenda as a simple rule. Conventional campaigns for elections involves cost, time, exertion for dispersing ones motivational opinion in order to get attention of voters, in contrary to this, social media campaign especially twitter involves no such cost. The politicians and political parties can outreach voters free of cost and in no time in deliberating party's agenda and ideology. The political parties can address users from all walks of life mostly involving younger generation by disseminating appealing agenda goals and objectives. While campaigning on twitter, political parties projects their positivity, in the meantime also propagates negativity of opponents. This sets a mobilization mechanism for voters and followers.

The political developments and events are reflected in tweets and even resulted in top trends maintained by twitter. German Federal Elections were held in 2009, Tumasjan revealed tweets obtained for related political parties, leader asserted resemblances with originally compiled results and concluded twitter can act as a mirror to offline political landscape. He predicted winner or loser for a candidate with accuracy of around 88.0importance of twitter and other social media applications was also unveiled during Singapore General Elections 2011 which found twitter to be integral part of election campaign and mobilization of citizens to cast

vote . The conventional Irish General Elections 2011 results were related by social analysts and researchers with combined approach involving volume based and sentiment based results obtained from twitter .

1.1.3 Twitter as Prediction Tool

In 2009 American Association of Public Opinion Researchers (AAPOR) spent $2billion for online research$, billion were spent for traditional forums, weblogs, and political discussion boards. The main focus of AAPOR was contents obtained from online surveys are beneficial and reliable rather than user generated contents. On the other hand Tamasjan worked around the point that Micro-blogging contents may be effective for opinion mining and predictions. His investigation proved that Micro-blogging content's information do base upon opinions of certain users or vetting of opinions of reliable users in their social networks . Thus their opinion on Micro-blogging forum may have certain weight which cannot be ignored and can be negotiated for opinion mining and predictions. Tamsjan further proved that accurate results and prediction for elections outcomes based on the data obtained from social media rather than political forums, and weblogs already supported by AAPOR. Social media also possess same features pertaining to data as of obtained from financial market containing aggregation mechanism from dispersed bits of information. Price system information can be referred to Micro-blogging twitter data by considering the size of followers and re-tweet rates, and most frequently used terms. These features influenced by human behavior can be used to predict currently occurring events . In addition to above mentioned salient features, detected sentiment also holds valuable data which be aggregated in to meaningful, predictable information. Previous studies as of US Elections, Singapore elections with extensive statistical outcomes proved twitter to be a social sensor for the prediction of electoral results which can be related to poll results.

1.2 Motivation

One such application is in the field of politics, where political entities need to understand public opinion and thus determine their campaigning strategy.

1.3 Objective

This approach is going to use the concept of Sentiment Analysis i.e. tracking opinion of public, which uses the natural language processing and extract the information like either public's view is positive or negative which can be used further to predict popularity of political party member

1.4 Report Organization

The project report is divided into four sections. Section 2 describes literature survey. Section 3 describes the methodology used for implementing the project. Section 4 gives the results and discussions. Finally Section 5 gives the conclusion.

Chapter 2

Literature Survey

There are many research works published in the area of Sentiment Analysis . The part of speech tagging is used in the Penn Treebank Project.

Pankaj Gupta, authors has used Naïve Bayes and SVM and created a collection of useful text from different Bag-Of-Words (BOW) and presented the summary. Rui Xia, they have presented textual dissection in real sense as: the 9 datasets, 2 antonym dictionaries and 3 classification algorithms; has been inspected and classified using binary classification into two classes (positive and negative). Moreover, the binary classes further extended into ternary classes (positive-negative-neutral) that includes third class as neutral reviews. The benefit is that, it is very operative for parting classification and extending the DSA algorithm. Md. Ansarul Haque and Tamjid Rahman has proposed analysis of sentiments using Fuzzy Logic where they use emotion analysis with the help of fuzzy logic that will help the creators or consumers or any concerned person for taking the actual decision according to their product or service interest. The added value is that it is supportive for anyone in any way to meet up their benefits or what they justify.

Minara P Antony, proposed about POS tagging method which is used to recognize the stop words and to distinct the sentiment terms. Unigram method is used to calculate the overall rating. The Stanford collection for organizing the data into negative, positive and neutral words and twofold prediction to properly identify the polarity of the data is being proposed. Rushlene Kaur Bakshi, discusses about emotion exploration which is a linguistic independent technology and also applied in the study of sociology, law, psychology etc. Agrawal, Rakesh, and Ramakrishnan Srikant discusses about some new hybrid fast algorithms for association

rule mining.

Authenticating and accessing of the Live Tweets is done by a Python Library, Tweepy. As Twitter requires all requests to use OAuth for Authentication and this Tweepy is an API that helps to authenticate a user and allows the user to access Live Twitter Data by just creating a Twitter application and retrieving your API access keys and tokens. These tokens will allow you to authenticate your Python client application with Twitter. TextBlob, a Python Library is used for noun phase extraction, POS Tagging, Classification, Tokenization, Word Inflection, Word and Phrase frequency, etc. NLTK is slightly different from TextBlob as TextBlob has a MIT License, is built on the top of NLTK and is very easily accessible. TextBlob is used for fast prototyping. On Comparing the Code Quality as calculated and provided by Lumnify, TextBlob is L3 and that of NLTK is L2. They vary from L1 to L5 with "L5" being the highest. TextBlob is used to provide an API for diving into common NL Processing tasks. Finally, the proposed code uses the above libraries to display the polarities of the Live Twitter Reviews as negative or positive and then display some Positive and Negative Reviews. This is not 100% accurate but is very useful in consideration of millions of tweets in every minute.

Chapter 3

Methodology

3.1 Introduction

This project was developed using Agile Development Model. The entire project was divided into three sprints. In the first first sprint, the characters for the password was developed. The designing of front-end and development of back-end was done in the second and third sprint respectively.

Natural Language Toolkit (NLTK), Tweepy and TextBlob are the most important tool the author has worked on.The flow of work is explained in the following modules.

A. Requirements

As the live Twitter Data is going to be accessed, the user need to import the Tweepy (Python Client for official Twitter API) and TextBlob.
For this, first the user need to install “pip”, a package used to install software packages written in python. After that the user needs to install Tweepy and TextBlob, using the following commands on the Terminal window:-
pip install tweepy
pip install textblob

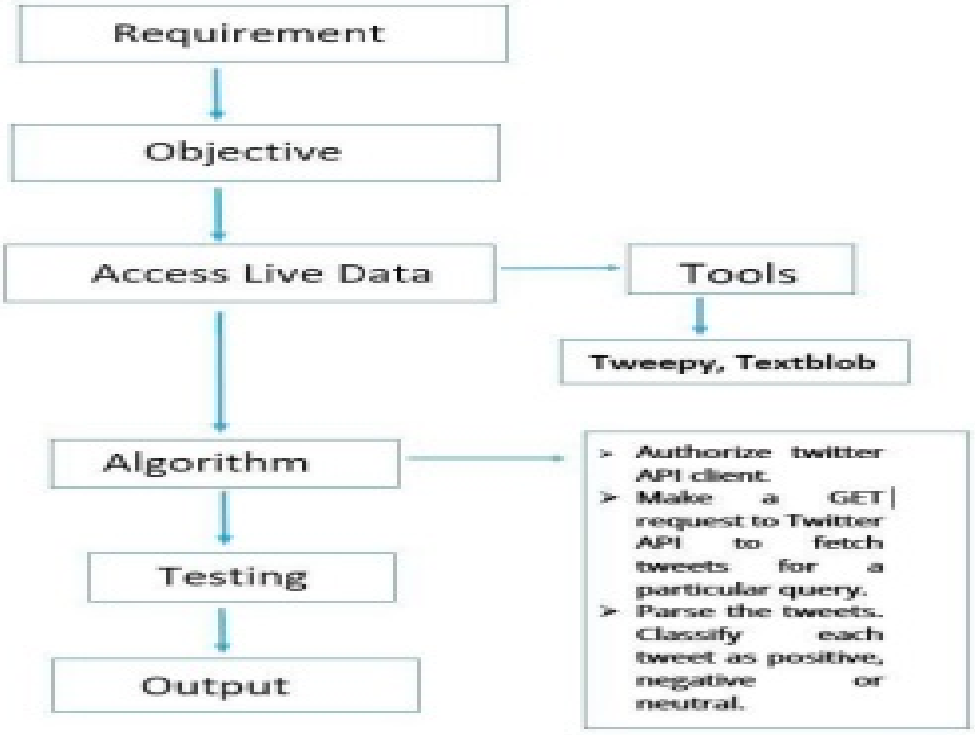

Figure 3.1: Flow Chart of Sentiment Analysis

B. Objective

Develop an algorithm that would take the query of the person's name for whom the user want to calculate the percentage of positive tweets and the percentage of negative tweets.

C. Access Live Data and Processing

I. Tweepy: - The users have to register their app on the Twitter website and then get the tweets. This is done at the Application Management of the Developers Section of Twitter. This is a very important step for the OAuth Authentication of Tweepy tool.
After creation of the application, the user need to generate private keys. These are required for OAuthHandlertakes this as the parameters. This is shown in dev.twitter.com. After successful generation of all the keys, we will copy the keys which will be further used in our Algorithm.

II. TextBlob: - The data in the text format is usually processed using the Textblob Library. TextBlob objects are treated as if they were like Python strings that learned to do the natural language processing

D. Algorithm Description

The algorithm proposed in our work has mainly 3 major steps.
• Authenticate twitter account.
• GET Request is made to the twitter for getting the tweets.
• Select the tweets. Segregate each tweet positive, negative.

The user has to make a twitter api-client. The given class contain the function which allow us to access the tweets. _init_ function is used for the authentication purpose. The clean tweets is used for cleaning the dataset acquired by the twitter api. For this the user has to import the Regular Expression library which is available in Python.
In get_tweets function, the user uses the following piece of code to call the API to get the tweets: -
fetched_tweets=self.api.search(q=query, count=count)

In get_tweet_sentiment function the user uses the TextBlob module:-
analysis = TextBlob(self.clean_tweet(tweet))

A classifier function divides the tweets as positive and negative polarity in the range of -1.0 to 1.0 Creation of Sentiment Classifier:-

* TextBlob contains a Movie dataset where positive and negative reviews has already been labelled.
* From each positive and negative review, positive and negative features are extracted respectively.
* This data with positive and negative features is now trained on Naive Bayes Classifier.

Then, the user can see a TextBlob class, where sentiment.polarity method can be used to get the polarity of tweets between -1 to 1.
Then, we classify polarity as: if analysis.sentiment.polarity is greater than 0 then return 'positive' else return 'negative'.
Finally, parse tweets are returned and then the user will calculate the percentage of positive and negative tweets.

E. Comparative Study
This code is using naïve Bayes for classification while there are several other methods present in these scenario to classify the Dataset.

3.2 Module

3.2.1 USER

*A user can register and login.
*After login user inputs a politician's name
*User can view the result

3.2.2 ADMIN

*Admin creates the user interface for the users

*Admin can view,delete and add new users

3.3 Algorithm

Naive Bayes classifier :

*In probabilistic learning, the probabilities associated with various events are used for learning and prediction.

*The probability is a number between 0 and 1, which indicates the chance that an event will occur.

*A probability of 0 indicates that there is no chance for the event to occur.

*A probability of 1 indicates that there is 100 percent chance for the event to occur.

*This learning model is used in Naive Bayes Algorithm.

*The technique used in this algorithm is based on the work by 18th Century Mathematician, Thomas Bayes.

*Naive Bayes algorithm is based on Bayes Theorem.Bayes theorem gives the conditional probability of an event A given another event B has occurred.

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B|A)P(A)}{P(B)}$$

- P(A|B) = Conditional probability of A given B
- P(B|A) = Conditional probability of B given A
- P(A) = Probability of event A
- P(B) = Probability of event B

Figure 3.2: Home

3.4 Developing Environment

3.4.1 Hardware Requirements

Processor: i3

Hard Disk: 500 GB

RAM: 4 GB

3.4.2 Software Requirements

Language: Python

Front End: Python-django

Back end: SQlite

Algorithm: Naive Bayes

IDE: Visual Studio Code

OS: Windows

Chapter 4

Results and Discussions

4.1 Results

Figure 4.1: Home

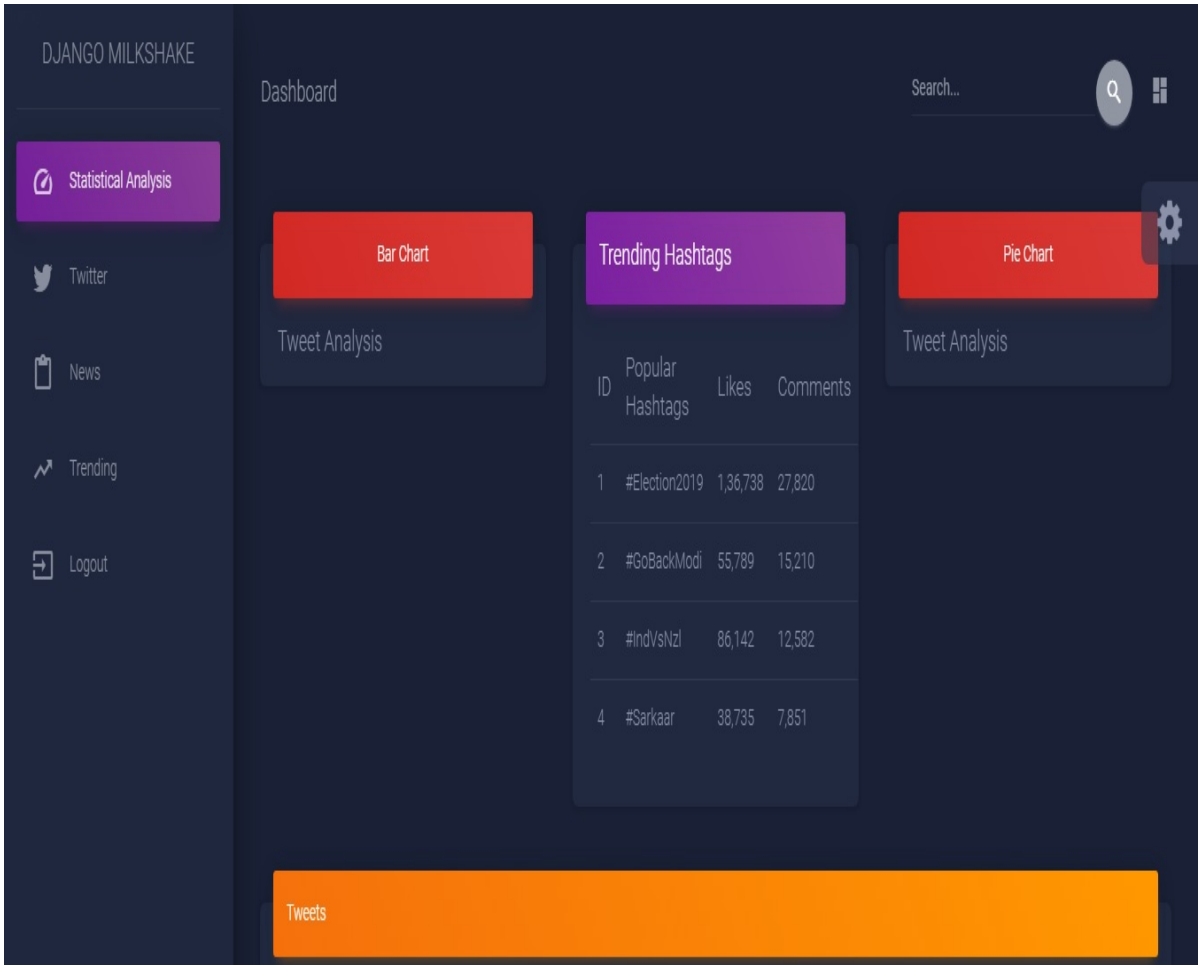

Figure 4.2: Dashboard

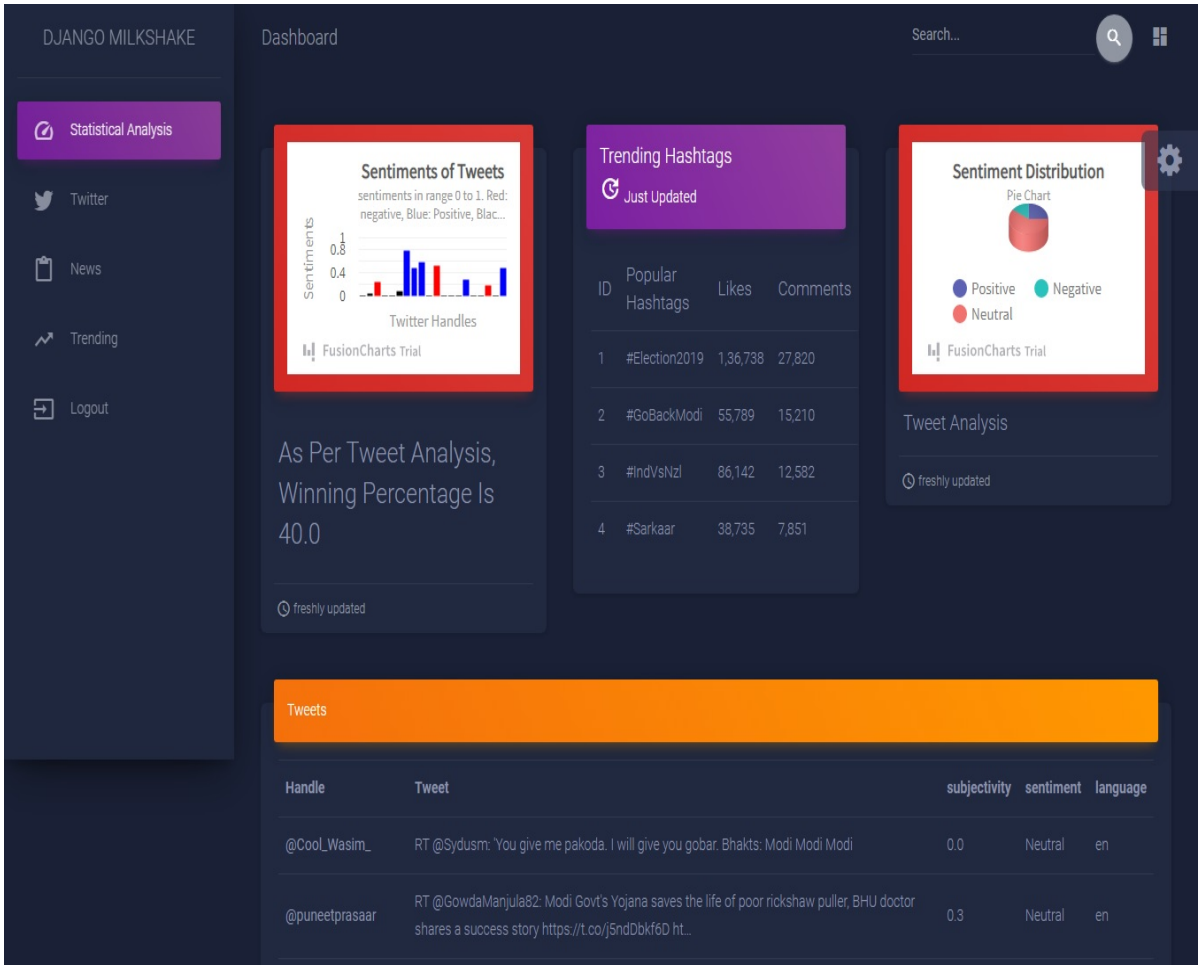

Figure 4.3: Result

Chapter 5

Conclusions

Sentiment detection / opinion mining from text obtained from social media has been appealing topic for Natural Language Processing. We have tried to find out the predictive power of social media. By using different classifiers we achieved around 70% accuracy for positive and negative sentiment. We also achieved around 50% accuracy for manual labeled data for three classes positive, negative and neutral. This work of forecasting elections for developing countries is of unique attempt. We have found social media user contributing for all political parties. We also deduced that there are certain political parties and leaders who have low electability but high popularity.

References

[1] **Yalin Yener** Senior GIS Analyst — Data Analyst — Data Scientist — Engineer. https://towardsdatascience.com/step-by-step-twitter-sentiment-analysis-in-python-d6f650ade58d.

[2] **Yogev Matalon, Ofir Magdaci, Adam Almozlino Dan Yamin** Published: 31 March 2021. https://www.nature.com/articles/s41598-021-86510-w.

[3] .https://www.youtube.com/watch?v=MA5nwEd9hKY.

[4] **Hafiz Syed Muhammad Bilal, Muhammad Asif Razzaq, Sungyoung Lee** Dept. of Computer Engineering, Kyung Hee University, Korea.Dept. of Computer Science, SEECS NUST, Pakistan. https://www.koreascience.or.kr/article/CFKO201435553770204.pdf.

Appendix

Source Code

Views.py

```
from django.shortcuts import render, redirect
# from django.contrib.auth import login, authenticate
from django.http import HttpResponse
import requests
import json
import tweepy
from tweepy.auth import OAuthHandler
from textblob import TextBlob
from .fusioncharts import FusionCharts
from collections import OrderedDict
from .news import Analysis

# Create your views here.


def home(request):
    return render(request, 'blog/index.html', {})


def getSentiments(request):
    auth = OAuthHandler('XRCnQ49KVWgy0IsN5QYBTn5Zm', 'P6UwYNbfboKQfrr51P3HLjp88Mq32SxNcQt7zsFKDdAZdXrAoW')
    auth.set_access_token('912853951984238592-BODZqgKvgD0QdKD5Rz1grMGPCDFiZm4',
          'proz3qXFAR7Ie8YOylG86z0uERL8orw0HcClAA2X4CN6t')

    api = tweepy.API(auth, wait_on_rate_limit=True, wait_on_rate_limit_notify=True)


    public_tweets = []
    if request.method=='GET':
        query = request.GET.get('search',"congress")
        a = Analysis(query)
        full_analysis = a.run()
    else:
        return redirect('home')

    chartData = OrderedDict()
    total_pos = 0
    total_neg = 0
    total_neu = 0

    for tweet in tweepy.Cursor(api.search, q=query, result_type="recent", tweet_mode="extended", lang="en").items(20):
        blob = TextBlob(tweet.full_text)
```

```
    senti = blob.sentiment.polarity
    sub = float("{0:.2f}".format(blob.sentiment.subjectivity))
    if senti>0.1:
        senti_analysis = 'Positive'
        total_pos += 1
    elif senti<-0.1:
        senti_analysis = 'Negative'
        total_neg += 1
    else:
        senti_analysis = 'Neutral'
        total_neu += 1
    public_tweets.append({'tweet':tweet, 'subjectivity':sub, 'senti': senti, 'senti_analysis':senti_analysis})
    chartData[tweet.user.screen_name] = senti
score=total_pos+(total_neu/2)-total_neg
percent=(score/20)*100
if(percent>95):
    percent=percent-10
print(percent)
dataSource = OrderedDict()

# The `chartConfig` dict contains key-value pairs data for chart attribute
chartConfig = OrderedDict()
chartConfig["caption"] = "Sentiments of Tweets"
chartConfig["subCaption"] = "sentiments in range 0 to 1. Red: negative, Blue: Positive, Black: Neutral"
chartConfig["xAxisName"] = "Twitter Handles"
chartConfig["yAxisName"] = "Sentiments"
chartConfig["numberSuffix"] = ""
chartConfig["theme"] = "fusion"

# The `chartData` dict contains key-value pairs data
dataSource["chart"] = chartConfig
dataSource["data"] = []

# Convert the data in the `chartData` array into a format that can be consumed by FusionCharts.
# The data for the chart should be in an array wherein each element of the array is a JSON object
# having the `label` and `value` as keys.

# Iterate through the data in `chartData` and insert in to the `dataSource['data']` list.
for key, value in chartData.items():
    data = {}
    data["label"] = key
    if value>0.1:
        data["color"] = "#0000ff"
    elif value<-0.1:
        data["color"] = "#ff0000"
    else:
        data["color"] = "#000000"
    if value>0:
        data["value"] = value
    else:
        data["value"] = -value
    dataSource["data"].append(data)

datapie = [{
        "label": "Positive",
        "value": total_pos
    }, {
        "label": "Negative",
        "value": total_neg
    }, {
        "label": "Neutral",
        "value": total_neu
    }]
```

Appendix

```
   # Create an object for the column 2D chart using the FusionCharts class constructor
   # The chart data is passed to the `dataSource` parameter.
   column2D = FusionCharts("column2d", "ex1" , "100%", "200", "chart-1", "json", dataSource)
      # Create an object for the pie3d chart using the FusionCharts class constructor
   pie3d = FusionCharts("pie3d", "ex2" , "100%", "200", "chart-2", "json",
      # The data is passed as a string in the `dataSource` as parameter.
   {
      "chart": {
         "caption": "Sentiment Distribution",
         "subCaption" : "Pie Chart",
         "showValues":"1",
         "showPercentInTooltip" : "1",
         "numberPrefix" : "$",
         "enableMultiSlicing":"1",
         "theme": "fusion"
      },
      "data": datapie,
   })

   # returning complete JavaScript and HTML code, which is used to generate chart in the browsers.
#  return render(request, 'index.html', {'output' : pie3d.render(), 'chartTitle': 'Pie 3D Chart'})
```

```
   return render(request, 'search_result.html', {'public_tweets': public_tweets, 'full_analysis': full_analysis,
        'percent':percent,'output1': column2D.render(), 'output2' : pie3d.render()})
```

search_result.html

```
{% extends 'base.html' %}
{% block title %}Tweets{% endblock %}
{% block js %}
   <script type="text/javascript" src="//cdn.fusioncharts.com/fusioncharts/latest/fusioncharts.js"></script>
   <script type="text/javascript"
        src="//cdn.fusioncharts.com/fusioncharts/latest/themes/fusioncharts.theme.fusion.js"></script>
{% endblock %}
   Twitter
   <ul>
      {% for tweet in public_tweets %}
         <li><b>@{{tweet.tweet.user.screen_name}}</b> - {{tweet.tweet.full_text}}<br>Sub:{{tweet.subjectivity}}
              Senti:{{tweet.senti}} ({{tweet.tweet.lang}})</li>
      {% endfor %}
   </ul>
   News
   <ul>
      {% for analysis in full_analysis %}
         <li><b>{{analysis.source}}</b> - {{analysis.text}}<br>{{analysis.sentiment}}</li>
      {% endfor %}
   </ul>
   Chart


{% block content %}
<div class="container-fluid">
      <div class="row">
         <div class="col-lg-4">
         <div class="card card-chart">
            <div class="card-header card-header-danger">
```

Appendix

```
    <div class="ct-chart" style="text-align: center">
       <div id="chart-1">{{ output1|safe }}</div>
    </div>
    </div>
    <div class="card-body">
    <h4 class="card-title"> <h4 class="card-title"><h3 color="red">As Per Tweet Analysis, Winning Percentage Is
         {{percent}}</h3></h4>
    </div>
    <div class="card-footer">
    <div class="stats">
       <i class="material-icons">access_time</i> freshly updated
    </div>
    </div>
</div>
</div>
<div class="col-lg-4">
<div class="card">
    <div class="card-header card-header-primary" id="trending">
    <h4 class="card-title">Trending Hashtags</h4>

    <i class="material-icons">update</i> Just Updated
    </div>
    <div class="card-body table-responsive">
    <table class="table table-hover">
       <thead class="text-warning">
       <th>ID</th>
       <th>Popular Hashtags</th>
       <th>Likes</th>
       <th>Comments</th>
       </thead>
       <tbody>
       <tr>
          <td>1</td>
          <td>#Election2019</td>
          <td>1,36,738</td>
          <td>27,820</td>
       </tr>
       <tr>
          <td>2</td>
          <td>#GoBackModi</td>
          <td>55,789</td>
          <td>15,210</td>
       </tr>
       <tr>
          <td>3</td>
          <td>#IndVsNzl</td>
          <td>86,142</td>
          <td>12,582</td>
       </tr>
       <tr>
          <td>4</td>
          <td>#Sarkaar</td>
          <td>38,735</td>
          <td>7,851</td>
       </tr>
       </tbody>
    </table>
    </div>
</div>
</div>
<div class="col-lg-4">
    <div class="card card-chart">
       <div class="card-header card-header-danger">
```

```
            <div class="ct-chart" style="text-align: center">
               <div id="chart-2">{{ output2|safe }}</div>
            </div>
            </div>
            <div class="card-body">
            <h4 class="card-title">Tweet Analysis</h4>
            </div>
            <div class="card-footer">
            <div class="stats">
               <i class="material-icons">access_time</i> freshly updated
            </div>
            </div>
         </div>
      </div>
   </div>
      <div class="row">
      <div class="col-12" id="twitter">
         <div class="card">
            <div class="card-header card-header-tabs card-header-warning">
               Tweets
            </div>
            <div class="card-body">
            <div class="tab-content">
               <div class="tab-pane active" id="profile">
               <table class="table">
                  <tbody>
                     <tr>
                        <th>Handle</th>
                        <th>Tweet</th>
                        <th>subjectivity</th>
                        <th>sentiment</th>
                        <th>language</th>
                     </tr>
                  {% for tweet in public_tweets %}
                     <tr>
                        <td><b>@{{tweet.tweet.user.screen_name}}</b></td>
                        <td>{{tweet.tweet.full_text}}</td>
                        <td>{{tweet.subjectivity}}</td>
                        <td>{{tweet.senti_analysis}}</td>
                        <td>{{tweet.tweet.lang}}</td>
                     </tr>
                  {% endfor %}
                     </div>
                  </tbody>
               </table>
            </div>
            </div>
         </div>
         </div>
      </div>
      <div class="col-12" id="news">
         <div class="card">
            <div class="card-header card-header-tabs card-header-success">
               News
            </div>
            <div class="card-body">
            <div class="tab-content">
               <div class="tab-pane active" id="profile">
               <table class="table">
                  <tbody>
                     <tr>
                        <th>Source, Time</th>
                        <th>News</th>
```

```
                        <th>Sentiment</th>
                    </tr>
                    {% for analysis in full_analysis %}
                        <tr>
                            <td><b>{{analysis.source}}</b></td>
                            <td>{{analysis.text}}</td>
                            <td>{{analysis.senti_analysis}}</td>
                        </tr>
                    {% endfor %}
                    </div>
                </tbody>
            </table>
        </div>
        </div>
    </div>
    </div>
</div>

{% endblock %}
```

User story ID	Priority <High/Medium/Low>	Size (Hours)	Sprint <#>	Status <Planned/In progress/Completed>	Release Date	Release Goal
1	Medium	5	1	Completed	27/12/2021	UI design
2	Medium	5		Completed	28/12/2021	User Registration, Login Form
3	Medium	10	2	Completed	22/01/2022	Collection of data
4	Medium	10	3	Completed	27/01/2022	Classification(Naïve Bayes)
5	Medium	10	4	Completed	06/02/2022	View prediction
6	Medium	10		Completed	12/02/2022	Generate Result

Figure A.1: Product Backlog

Backlog Item	Status and Completion date	Original Estimate in hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12
User story #1,2			Hours	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs
UI Designing	27/12/21	5	5	0	0	0	0	0	0	0	0	0	0	0
Coding	28/12/21	5	0	5	0	0	0	0	0	0	0	0	0	0
User story #3														
Data Collection	22/01/22	10	0	0	4	4	3	0	0	0	0	0	0	0
User story #4														
Classification	27/01/22	10	0	0	0	0	0	5	5	0	0	0	0	0
User story #5,6														
Prediction	06/02/22	10	0	0	0	0	0	0	0	4	3	3	0	0
Generate Result	12/02/22	10	0	0	0	0	0	0	0	0	0	0	5	5
Total		50	5	5	4	4	3	5	5	4	3	3	5	5

Figure A.2: Sprint Plan

Backlog Item	Status and Completion date	Original Estimate in hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Completed <Y/N>
User story #1,2			Hours	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	
UI Designing	27/12/21	5	5	0	0	0	0	0	0	0	0	0	0	0	Y
Coding	28/12/21	5	0	5	0	0	0	0	0	0	0	0	0	0	Y
User story #3															
Data Collection	22/01/22	10	0	0	4	4	3	0	0	0	0	0	0	0	Y
User story #4															Y
Classification	27/01/22	10	0	0	0	0	0	5	5	0	0	0	0	0	
User story #5,6															Y
Prediction	06/02/22	10	0	0	0	0	0	0	0	4	3	3	0	0	
Generate Result	12/02/22	10	0	0	0	0	0	0	0	0	0	0	5	5	Y
Total		50	5	5	4	4	3	5	5	4	3	3	5	5	

Figure A.3: Sprint Actuals